NOTE SULLO SVOLGIMENTO DELLA PROVA SCRITTA:

- **SPEGNERE I TELEFONINI;**
- **SCRIVERE IL PROPRIO NOME, COGNOME E NUMERO DI MATRICOLA SU OGNI FOGLIO UTILIZZATO;**
- **NON È POSSIBILE CONSULTARE NESSUN TIPO DI MATERIALE;**
- **NON È POSSIBILE UTILIZZARE CALCOLATRICI;**
- **PRIMA DI SCRIVERE LA SOLUZIONE DELL'ESERCIZIO, INSERIRE IL NUMERO DI ESERCIZIO CHE SI STA RISOLVENDO. PER ESEMPIO, SCRIVERE "ESERCIZIO N. 1" QUANDO SI STA RISOLVENDO L'ESERCIZIO N. 1;**
- NON COPIARE DAL VICINO (NON È DETTO CHE IL VICINO SIA PIÙ BRAVO DI VOI);
- NON PERMETTETE AL VICINO DI COPIARE (È SPIACEVOLE VEDERSI ANNULLARE IL COMPITO SENZA COLPE);
- **I PRIMI TRE ESERCIZI VALGONO 6 PUNTI. I RIMANENTI TRE VALGONO 4 PUNTI.**
- **ALLA FINE DELLA PROVA, RICONSEGNARE TUTTI I FOGLI UTILIZZATI.**
- **SI RICORDA CHE LO SCRITTO VALE PER TRE APPELLI CONSECUTIVI (COMPRESO L'APPELLO IN CUI LO SCRITTO È SOSTENUTO)**

1) Scrivere una funzione ricorsiva che, dato un vettore di caratteri, ricerca nel vettore il carattere passato come argomento alla funzione e ne restituisce l'indice della prima occorrenza. Se il carattere non è presente, la funzione restituisce –1. La funzione deve poter essere chiamata con vettori di dimensione qualsiasi. Sia dato il vettore:

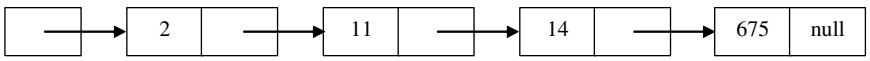
   v=['q', 'e', 's', 'a', 's', 'z']

   Se, per esempio, viene cercato il carattere 's', la funzione restituisce 2.

2) Sia data la struttura seguente

   ```
   struct elem{int info; elem* pun;};
   ```

   Scrivere una funzione che costruisce una lista di elementi di tipo elem, ordinata per valori crescenti del campo info. I valori degli elementi della lista vengono letti dal file input.txt. Assumere che il file contenga solo interi separati da spazi bianchi. Per esempio, se il file input.txt contiene:
   14 675 2 11
   la funzione restituisce la lista seguente:



3) Scrivere una funzione che, dato un numero intero n>0, restituisce un vettore v di dimensione n tale che il generico elemento v[k] punta ad una stringa composta dalla sequenza ciclica delle prime k+1 lettere minuscole dell'alfabeto. Se n = 28,

   v[0] = "a"
   v[1] = "ab"
   ....
   v[25] = "abc.....z"
   v[26] = "abc.....za"
   v[27] = "abc.....zab"

4) Data la rappresentazione $(11011110110)_2$ in base 2, trasformarla in base 8.
   Scrivere la rappresentazione in complemento a due su 8 bit del numero intero -105.

5) Si mostri l'uscita a video del programma C++ seguente:

```
#include <iostream>
using namespace std;

class A
{ protected:
      int x;
  public:
      A(int n=0) : x(n) { cout << "nuovo A: x = " << x << endl;}
      ~A() { cout << "via A" << endl;}
};

class B: public A
{   A a;
    int y;
  public:
      B(int n=5): A(n) { y=n+x; cout << "nuovo B: y = " << y << endl; }
      virtual ~B() { cout << "via B" << endl;}
};

class C: public B
{
public:
     C(int k=0) : B(k) { x++;  cout << "nuovo C: x = " << x << endl; }
     ~C() { cout << "via C" << endl;}
} ;
```

```
int main(){
    A* pa = new C(6);
    B* pb = new C;
    delete pa;
    delete pb;
    return 0;
}
```

6) Si mostri l'uscita a video del programma C++ seguente con ingresso -1, 0, +1:

```
#include<iostream>
using namespace std;
class A {
  public:
   virtual void m1() {cout << "A::m1()" << endl;}
   void m2(){cout << "A::m2()" << endl;;}
};

class B : public A {
  public:
   virtual void m1() {cout << "B::m1()" << endl;}
   void m2(){cout << "B::m2()" << endl;}
};


void g(int x) {
   if (x<0) throw B();
   if (x==0) throw A();
   throw 'a';
   cout << x + 30 << endl;
}
```

```
void f(int x) {
     try
     { g(x); }
     catch(A& a){ a.m1(); a.m2();}
     cout << x + 50 << endl;
}

int main () {
   int x;
   cin >> x;
   try{ f(x);}
   catch(...){cout <<  "catch generico " << endl;}
   cout << x + 100 << endl;
   return 0;
   }
```